

KOREAN PATENT ABSTRACT (KR)

Patent Laid-Open Gazette

(51) IPC Code: B41J 2/435

(11) Publication No.: P1999-0049428

(43) Publication Date: July 5, 1999

(21) Application No.: 10-1997-0068368

(22) Application Date: December 12, 1997

(71) Applicant:

LG Electronics Co., Ltd.

20, Yeouido-dong, Yeongdeungpo-gu, Seoul, Korea

(72) Inventor:

KIM, BYUNG HAN

(54) Title of the Invention:

Controller Circuit of Laser Printer

Abstract:

Provided is a controller circuit of a laser printer for reducing the number of EPROMs storing controller programs, and improving operational speed of the controller program. In a general controller circuit of a laser printer, it takes a time of two-cycles (62.5ns × 2 = 125ns) for a CPU to access the EPROM and read program, thus an operational speed of performing the program is lowered. In addition, since a plurality of EPROMs of high prices should be required to store the controller programs, costs for configuring the circuit increase. However, according to the present invention, since the controller program is compressed to be stored, the number of EPROMs can be reduced. Then the compressed controller program is recovered and stored in a high speed SRAM, and the controller program is performed in the high speed SRAM region, thereby improving the operational speed of the controller program.

특 1999-0049428

(19) 대한민국특허청(KR)  
 (12) 공개특허공보(A)

(51) Int. Cl. B41J 2/435	(11) 공개번호 특 1999-0049428
(21) 출원번호 10-1997-0068368	(43) 공개일자 1999년 07월 05일
(22) 출원일자 1997년 12월 12일	
(71) 출원인 엘지전자 주식회사 구자홍	
(72) 발명자 김병한	서울특별시 영등포구 여의도동 20번지
(74) 대리인 김영철	경기도 평택시 승단지역 지산동 1094 건영마파트 101동 804호
<b>설사성구 : 없음</b>	
<b>(54) 레이저 프린터의 콘트롤러 회로</b>	

**요약**

본 발명은 레이저 프린터의 콘트롤러 회로에 관한 것으로서, 특히, 콘트롤러 프로그램을 저장하기 위한 EEPROM의 갯수를 감소시킴과 동시에 콘트롤러 프로그램의 실행속도를 향상 시키도록 하는 레이저 프린터의 콘트롤러 회로에 관한 것이다.

종래의 레이저 프린터의 콘트롤러 회로에서 CPU가 EEPROM을 액세스하여 프로그램을 읽어들이는 경우 2사이클 시간( $62.5\text{ns} \times 2 = 125\text{ns}$ )이 소요되므로 EEPROM의 프로그램을 읽어들임에 있어 장시간이 소요되어 프로그램 실행속도가 떨어지고, 콘트롤러 프로그램을 저장하기 위하여 고가의 EEPROM을 여러개 구비해야 하므로 회로구성을 경제성 있게 할 수 없다는 문제점이 있다.

본 발명은 콘트롤러 프로그램을 압축하여 저장하므로 EEPROM을 절감하고, 압축된 콘트롤러 프로그램을 복원하여 고속 SRAM에 저장하고 고속 SRAM영역에서 콘트롤러 프로그램을 실행하므로 콘트롤러 프로그램의 실행속도를 향상시키게 된다.

**도표도**

도2

**설명서****도면의 간략한 설명**

도1은 종래 레이저 프린터의 콘트롤러 회로 구성도

도2는 본 발명에 따른 레이저 프린터의 콘트롤러 회로 구성도

**\* 도면의 주요부분에 대한 부호의 설명 \***

10, 20: CPU 11, 12, 21: EEPROM

22a~22d: 고속 SRAM

**발명의 상세한 설명****발명의 목적****발명이 속하는 기술 및 그 분야의 종래기술**

본 발명은 레이저 프린터의 콘트롤러 회로에 관한 것으로서, 특히, 콘트롤러 프로그램을 저장하기 위한 EEPROM의 갯수를 감소시킴과 동시에 콘트롤러 프로그램의 실행속도를 향상시키도록 하는 레이저 프린터의 콘트롤러 회로에 관한 것이다.

일반적으로 레이저 프린터는 컴퓨터 시스템으로부터 인가되는 데이터를 수신하여 시각적으로 확인 할 수 있도록 용지에 인쇄하는 기능을 수행하는데 이와 같은 동작을 제어하기 위하여 자체에 콘트롤러 회로를 구비하고 있다.

종래의 레이저 프린터의 콘트롤러 회로는 첨부된 도1에 도시되어 있는 바와 같이 CPU(10)와 다수의 EEPROM(11, 12)로 구비하여 구성된다.

EEPROM(11, 12)에는 레이저 프린터의 제반동작을 제어하기 위한 프로그램이 저장되어있고, CPU(10)는 EEPROM(11, 12)에 저장된 프로그램을 읽어들여 실행함으로써 레이저 프린터의 여러 동작을 제어한다.

CPU(10)는 EPROM(11,12)을 익세스하는 경우 EPROM(11,12)에 대하여 칩선택 신호(CS\*), 출력 인에이블 신호(OE\*) 및 어드레스를 출력하여 EPROM(11,12)을 익세스하여 프로그램을 읽어 들인다.

이상과 같이 구성된 종래 레이저 프린터의 콘트롤러 회로에서는 EPROM에 대한 CPU(10)의 익세스 소요시간과 EPROM(11,12)의 데이터 출력 소요시간이 상이한데, CPU(10)가 메모리 수단을 익세스 하는데 소요되는 1사이클 시간은 62.5ns(CPU의 외부클럭이 16MHz인 경우를 기준으로 했을 때)이고, EPROM(11,12)이 익세스를 받았을 때 데이터를 출력할 때 있어 120ns가 소요된다. 즉, CPU(10)가 EPROM(11,12)을 익세스 하여 프로그램을 읽어 들이는 경우 2사이클 시간(62.5ns\*2=125ns)이 소요되므로, EPROM(11,12)의 프로그램을 읽어 들임에 있어서 장시간이 소요되어 프로그램 실행 속도가 떨어지는 문제점이 있다.

또한, 상술한 종래 레이저 프린터의 콘트롤러 회로에서는 콘트롤러 프로그램을 저장하기 위하여 고가의 EPROM(11,12)을 여러개 구비해야 하므로 회로 구성은 경제성 있게 할 수 없다는 문제점도 있다.

#### 발명의 이루고자 하는 기술적 과정

본 발명은 전술한 바와 같은 문제점을 감안하여 안출한 것으로, 콘트롤러 프로그램을 압축하여 저장함으로써 EPROM을 절감하고, 압축된 콘트롤러 프로그램을 고속 SRAM으로 복원하여 콘트롤러 프로그램을 실행함으로써 콘트롤러 프로그램의 실행속도를 향상시킴과 동시에 회로구성을 경제성 있게 할하도록 함에 목적이 있다.

#### 발명의 구성 및 작동

상기와 같은 목적을 달성하기 위한 본 발명의 특징은, 레이저 프린터의 콘트롤러 회로에 있어서, 레이저 프린터의 동작을 제어하기 위한 프로그램을 압축하여 저장하는 EPROM과; 압축상태로 부터 복원된 프로그램을 저장하는 고속 SRAM과; 상기 EPROM을 익세스 하여 EPROM의 압축저장된 프로그램을 읽어 들여 복원하고, 해당 복원된 프로그램을 상기 고속 SRAM에 저장한 후 상기 고속 SRAM을 익세스하여 프로그램을 실행하는 CPU를 포함하는 데 있다.

이하, 첨부된 도면을 참조하여 본 발명의 실시 예를 상세하게 설명하면 다음과 같다.

본 발명에 따른 레이저 프린터의 콘트롤러 회로는 첨부된 도면 도2에 도시한 바와 같이 CPU(20), EPROM(21), 고속 SRAM(22a-22d)를 구비하여 구성되어 있다. EPROM(21)에는 레이저 프린터의 제반 동작을 제어하기 위한 프로그램이 압축되어 저장되어 있다.

CPU(20)는 EPROM(21)에 압축 저장된 프로그램을 읽어 들여 원상태로 복원시켜 고속 SRAM(22a-22d)에 저장해 놓고, 고속 SRAM(22a-22d)을 익세스하면서 프로그램을 실행하여 제반 제어 동작을 수행한다. 또한, 고속 SRAM(22a-22d)은 CPU(20)로부터 인가 받은 프로그램을 저장하고, CPU(20)의 요구에 따라 프로그램을 CPU(20)에게 제공한다.

한편, CPU(20)는 EPROM(21)을 익세스 하는 경우에 EPROM(21)에 대하여 칩선택 신호(CS\*), 출력 인에이블 신호(OE\*) 및 어드레스를 출력하여 EPROM(21)을 익세스하여 압축저장된 프로그램을 읽어 들이는데, EPROM(21)에는 하드웨어 테스트를 위한 기본적인 프로그램은 압축되지 않은 형태로 저장하고 있고 그 외의 프로그램은 압축된 형태로 저장하고 있다. 그리고, CPU(20)는 고속 SRAM(22a-22d)을 익세스 하는 경우에 고속 SRAM(22a-22d)에 대하여 칩선택 신호(CS\*), 출력 인에이블 신호(OE\*) 및 어드레스를 출력하여 고속 SRAM(22a-22d)을 익세스 하여 고속 SRAM(22a-22d)에 프로그램을 저장하고, 고속 SRAM(22a-22d)의 프로그램을 실행하여 제반 제어동작을 수행한다. 또한, EPROM(21)이 익세스를 받았을 때 데이터를 출력함에 있어서 120ns의 시간이 소요되는 반면 고속 SRAM(22a-22d)이 익세스를 받았을 때 데이터를 출력함에 있어서 20ns의 시간이 소요되고, CPU(20)가 메모리 수단을 익세스하는데 소요되는 1사이클 시간은 62.5ns(어이의 외부클럭이 16MHz인 경우를 기준으로 했을 때)이다. 즉, CPU(20)가 EPROM(21)을 익세스하여 프로그램을 읽어 들이는 경우에는 2사이클 시간(62.5ns\*2=125ns)이 소요되나, 고속 SRAM(22a-22d)을 익세스하여 프로그램을 기록/판독하는 경우에는 1사이클 시간이 소요된다. 따라서, CPU(20)는 EPROM(21)에 압축 저장된 프로그램을 읽어 들여 복원해서 고속 SRAM(22a-22d)에 저장한 후 고속 SRAM(22a-22d)을 익세스하여 프로그램을 실행하므로, 프로그램 실행 속도가 향상되어 레이저 프린터의 제반 제어 동작을 신속히 행할 수 있게 된다.

전술한 바와 같이 구성되는 본 발명에 따른 레이저 프린터 콘트롤러 회로의 동작을 첨부된 도면에 따라 상세하게 설명하면 다음과 같다.

먼저, CPU(20)에 리셋신호가 인가되면 CPU(20)는 EPROM(21)을 익세스하여 EPROM(21)에 저장된 프로그램을 읽어 들여 처리하는데, EPROM(21)에는 기본적인 하드웨어 테스트를 담당하는 콘트롤러 프로그램은 압축저장되어 있지 않고, 그 외의 프로그램은 압축된 형태로 저장되어 있다. CPU(20)는 리셋되는 경우에, 먼저 EPROM(21)으로부터 하드웨어 테스트를 위한 프로그램을 읽어 들여 실행함으로써 하드웨어에 대한 테스트를 수행하고, 압축저장된 프로그램을 읽어 들여 복원하여 고속 SRAM(22a-22d)에 저장한 후 해당 고속 SRAM(22a-22d)을 고속으로 익세스하여 고속 SRAM(22a-22d) 영역에서 실행함으로써 콘트롤러 프로그램의 실행 속도를 향상시키게 된다. 익세스 시간이 120ns인 EPROM(21)에 저장된 데이터를 처리하는데 2사이클이 필요한 반면 익세스 시간이 20ns인 고속 SRAM(22a-22d)에 저장된 데이터를 처리하는데 1사이클이 필요하므로 실행 속도를 향상시키게 된다.

또한 프로그램을 EPROM(21)에 압축된 형태로 저장하므로 프로그램 저장영역을 감소 시킬 수 있어, EPROM(21)의 갯수를 대폭 절감시키게 된다.

#### 발명의 효과

전술한 바와 같이, 본 발명은 콘트롤러 프로그램을 압축하여 저장하므로 EPROM을 절감하고, 압축된 콘트롤러 프로그램을 복원하여 고속 SRAM에 저장하고 고속 SRAM 영역에서 콘트롤러 프로그램을 실행하므로 콘

트롤러 프로그램의 실행속도를 향상시키게 된다.

#### (5) 용구의 특성

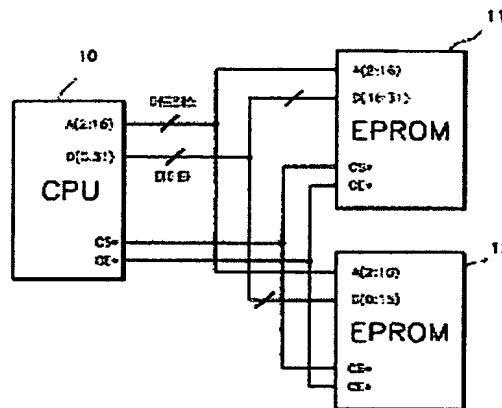
용구항 1. 레이저 프린터의 콘트롤러 회로에 있어서, 레이저 프린터의 동작을 제어하기 위한 프로그램을 암축하여 저장하는 EPROM과; 암축상태로 부터 복원된 프로그램을 저장하는 고속 SRAM과; 상기 EPROM을 액세스 하여 EPROM의 암축저장된 프로그램을 읽어 들여 복원하고, 해당 복원된 프로그램을 상기 고속 SRAM에 저장한 후 상기 고속 SRAM을 액세스하여 프로그램을 실행하는 CPU를 포함하는 것을 특징으로 하는 레이저 프린터의 콘트롤러 회로.

용구항 2. 제1항에 있어서

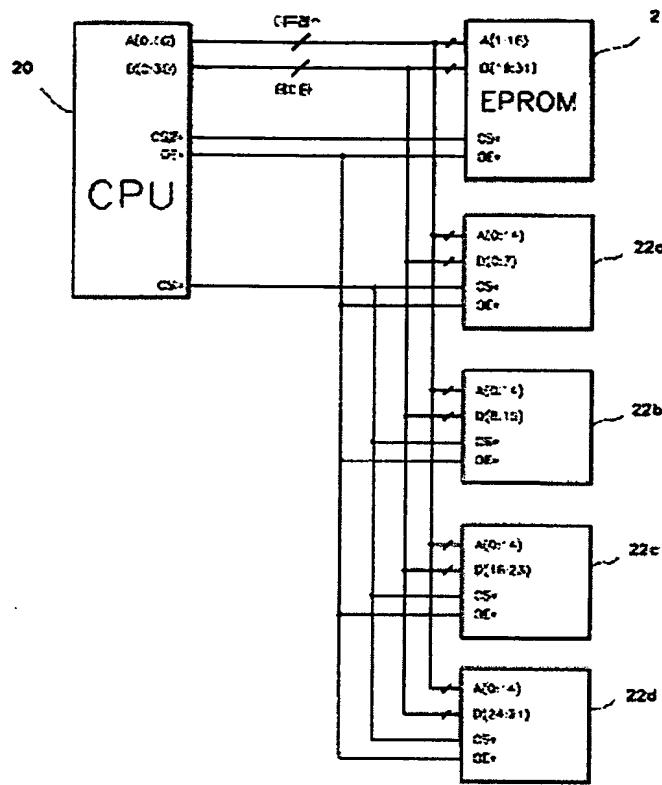
상기 고속 SRAM에 대한 액세스 속도는 상기 EPROM에 대한 액세스 속도 보다 빠른 것을 특징으로 하는 레이저 프린터의 콘트롤러 회로.

#### 도면

##### 도면 1



582



(10)日本国特許庁 (JP)

## (12) 公開特許公報 (A)

(11)特許出願公開番号

特開平8-234939

(43)公開日 平成8年(1996)9月13日

(51) Int.Cl.  
G 06 F 3/12  
B 41 J 5/30

識別記号 戸内整理番号

F I  
G 06 F 3/12  
B 41 J 5/30

技術表示箇所  
C  
Z

審査請求 未請求 請求項の数6 OL (全8頁)

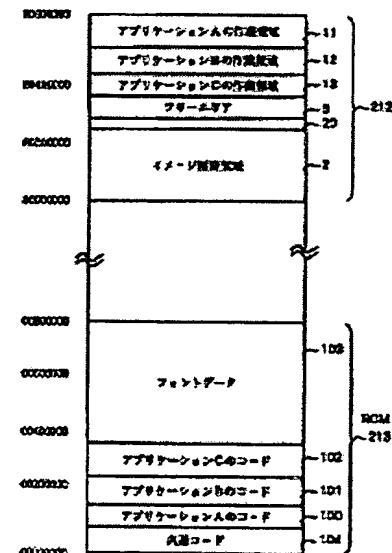
(21)出願番号 特願平7-40111  
(22)出願日 平成7年(1995)2月28日

(71)出願人 000001007  
キヤノン株式会社  
東京都大田区下丸子3丁目30番2号  
(72)発明者 庄司 雄之  
東京都大田区下丸子3丁目30番2号 キヤ  
ノン株式会社内  
(74)代理人 弁理士 大槻 康徳 (外1名)

(54) [発明の名称] 印刷装置及び該装置におけるデータ管理方法  
(61) [要約]

【目的】メモリ上のデータを効率良く圧縮できる印刷装置及び該装置におけるデータ管理方法を提供することを目的とする。

【構成】外部機器よりの印刷コマンドに応じてアプリケーションを切り換えて印刷処理を実行するプリント装置であって、ROM 213に複数のアプリケーション・プログラムを記憶しており、これら複数のアプリケーション・プログラムのそれぞれに對応して、RAM 213に、対応するアプリケーションで使用されるデータの属性に応じて連続してデータを配置する。そして印刷コマンドにより指示されたアプリケーションがその時点で実行中か否かを判断し、当該アプリケーションが実行中でない時は、その時点で実行中のアプリケーションに対応するワークエリアのデータを圧縮し、記憶されている複数のアプリケーション・プログラムのうち、その印刷コマンドにより指示されたアプリケーションに処理を移行して印刷する。



### 【特許請求の範囲】

【請求項1】 外部機器よりの印刷コマンドに応じてアプリケーションを切り換えて印刷処理を実行する印刷装置であつて、複数のアプリケーション・プログラムを記憶する記憶手段と、前記複数のアプリケーション・プログラムのそれぞれに対応して設けられ、対応するアプリケーションで使用されるデータの属性に応じて連続してデータを記憶するワークエリアと、前記記憶手段に記憶されているアプリケーション・プログラムのうち、前記印刷コマンドにより指示されたアプリケーションに処理を移行する制御手段と、前記制御手段によるアプリケーションの移行に際して、前記ワークエリアのデータを圧縮する圧縮手段と、を有することを特徴とする印刷装置。

【請求項2】 前記印刷コマンドにより指示されたアプリケーションは、当該アプリケーションの実行時、前記圧縮手段によるデータ圧縮により拡大された前記ワークエリアの使用可能エリアを使用することを特徴とする請求項1に記載の印刷装置。

【請求項3】 前記圧縮手段は、前記印刷コマンドにより指示されたアプリケーションがその時点で実行中か否かを判断し、当該アプリケーションが実行中でない時は、その時点で実行中のアプリケーションに対応するワークエリアのデータを圧縮することを特徴とする請求項1に記載の印刷装置。

【請求項4】 外部機器よりの印刷コマンドに応じてアプリケーションを切り換えて印刷処理を実行する印刷装置におけるデータ管理方法であつて、実行中のアプリケーションにより使用されるデータを、その属性に応じて連続してメモリに記憶する工程と、印刷コマンドにより指示されたアプリケーションに移行する際、その時点で実行中のアプリケーションに対応する前記メモリのデータを圧縮する圧縮工程と、を有することを特徴とする印刷装置におけるデータ管理方法。

【請求項5】 前記圧縮工程は、前記印刷コマンドにより指示されたアプリケーションがその時点で実行中であるか否かを判断し、実行中でない時にそれまで実行されていたアプリケーションに対応する前記メモリのデータを圧縮することを特徴とする請求項4に記載の印刷装置におけるデータ管理方法。

【請求項6】 前記印刷コマンドにより指示されたアプリケーションがその時点で実行中でない時は、記憶されているアプリケーション・プログラムのうち前記印刷コマンドにより指示されたアプリケーションに処理を移行する工程を更に有することを特徴とする請求項4に記載の印刷装置におけるデータ管理方法。

【産業上の利用分野】 本発明は、外部機器よりの印刷コマンドを受信し、該印刷コマンドを解釈して実行するアプリケーションに切り換えて印刷する印刷装置及び該装置におけるデータ管理方法に関するものである。

### 【0002】

【従来の技術】 近年、急速に普及してきたコンピュータ機器に接続され、そのコンピュータ機器よりの印刷データを受信して印刷する各種プリンタ装置が存在している。これらプリンタ装置のうち、初期のプリンタ装置のように、文字コードを受信して一定サイズの文字のみを印刷するプリンタ装置の場合には、プリンタ装置の主たる機能は、その送られてきた文字コードを解釈して文字パターンに展開するだけで十分であった。しかし、近年のように、文字の書体やサイズを変更したり、イメージの印刷を行うに伴い、プリンタ装置の機能に対する要求が次第に複雑・多様化している。これに伴って、コンピュータ機器よりプリンタ装置に送られるコマンドや印刷データ等が複雑になり、プリンタ装置では、これら種々のコマンドが組込まれた印刷コマンドを入力し、それを解釈する必要が生じてきた。

【0003】 しかし、プリンタ装置で使用されるコマンド体系は、そのメカ、機器別に独自に開発されているため、相互に互換性がないのが現状である。またコンピュータ機器（ホスト）のソフトウェアも、このような複数の印刷コマンドセットに対応しているが、全ての印刷コマンドセットに対応できるわけではない。そこでプリンタ装置においても、複数の印刷コマンドセットを組み込み、どのようなコンピュータ機器からでも印刷出力ができるように対応する必要が生じている。

【0004】 また一般に、プリンタ装置は、複数のコンピュータ機器から印刷コマンドを供給されることが多いため、それぞれの印刷コマンドを解釈するアプリケーションを、プリンタ装置を再起動することなく立ち上げられなければ、操作性が極めて悪いものとなる。なぜならば、アプリケーションの起動時には、時間をかけて作業領域に文字情報などを展開するため、そのアプリケーションの切り換えに多くの時間を要する。このようにプリンタ装置は、複数のアプリケーションを換時に切り替えられなければならない。

【0005】 プリンタ装置の制御部で、独立したメモリ等の資源をそれぞれのアプリケーションに割り当てられれば、プリンタ装置を再起動することなくアプリケーションを容易に換時に切り換えることができるが、それでは単独のアプリケーションが動作するプリンタ装置の複数台分のメモリ等の資源が必要となり、コスト的に現実的でない。

### 【0006】

【発明が解決しようとする課題】 そこで、メモリ上のアプリケーションの作業領域を一時退避させ、メモリを異なったアプリケーション間で共有する技術が必要とな

### 【発明の詳細な説明】

【0001】

る。コンピュータ機器などでは、RAM上の作業領域を一時的にハードディスク等の二次記憶領域に退避させた後、他のアプリケーションを立ち上げることにより、アプリケーションの切換えを高速に行うことが可能であるが、プリンタ装置の場合は、コストの点からこのような二次記憶装置を備えることは難しい。

【0007】本発明は上記從来例に鑑みてなされたもので、メモリ上のデータを効率良く圧縮できる印刷装置及び該装置におけるデータ管理方法を提供することを目的とする。

【0008】また本発明の目的は、外部機器よりの印刷コマンドに応じて、印刷装置内で実行するアプリケーションを高速に切り換えて印刷を行うことができる印刷装置及び該装置におけるデータ管理方法を提供することにある。

【0009】また本発明の他の目的は、データを効率良く圧縮して、アプリケーションを実行するためのワークエリアを十分に確保できるようにした印刷装置及び該装置におけるデータ管理方法に関するものである。

【0010】

【課題を解決するための手段】上記目的を達成するためには本発明の印刷装置は以下のような構成を備える。即ち、外部機器よりの印刷コマンドに応じてアプリケーションを切り換えて印刷処理を実行する印刷装置であつて、複数のアプリケーション・プログラムを記憶する記憶手段と、前記複数のアプリケーション・プログラムのそれぞれに対応して設けられ、対応するアプリケーションで使用されるデータの属性に応じて連続してデータを配置するワークエリアと、前記記憶手段に記憶されているアプリケーション・プログラムのうち、前記印刷コマンドにより指示されたアプリケーションに処理を移行する制御手段と、前記制御手段によるアプリケーションの移行に際して、前記ワークエリアのデータを圧縮する圧縮手段とを有する。

【0011】上記目的を達成するために本発明の印刷装置におけるデータ管理方法は以下のようない工程を備える。即ち、外部機器よりの印刷コマンドに応じてアプリケーションを切り換えて印刷処理を実行する印刷装置におけるデータ管理方法であつて、実行中のアプリケーションにより使用されるデータを、その属性に応じて連続してメモリに配置する工程と、印刷コマンドにより指示されたアプリケーションに移行する際、その時点で実行中のアプリケーションに対応する前記メモリのデータを圧縮する圧縮工程とを有する。

【0012】

【作用】以上の構成において、複数のアプリケーション・プログラムを記憶手段に記憶しておき、これら複数のアプリケーション・プログラムのそれぞれに対応して設けられ、対応するアプリケーションで使用されるデータの属性に応じて連続してデータをワークエリアに記憶す

る。記憶手段に記憶されている複数のアプリケーション・プログラムのうち、その実行した印刷コマンドにより指示されたアプリケーションに処理を移行する際、ワークエリアに記憶されているデータを圧縮するように動作する。

【0013】また本発明のデータ管理方法によれば、実行中のアプリケーションにより使用されるデータを、その属性に応じて連続してメモリに配置し、その印刷コマンドにより指示されたアプリケーションに移行する際、その時点で実行中のアプリケーションに対応するメモリのデータを圧縮する。

【0014】

【実施例】以下、添付図面を参照して本発明の好適な実施例を説明する前に、本発明の一実施例の処理の概要を簡単に説明する。

【0015】データを圧縮する場合、圧縮の対象になるデータが全くランダムに並んでいる場合と比較して、相間の高いデータが連続して並んでいる方が高い圧縮効率が期待できる。一般的には、RAMの作業領域には、ソフトウェアにより使用される様々なデータ形式のデータが無秩序に混在している。これをデータのタイプ毎に文字列形式、整数、アドレス情報等に分離して、同じデータ属性のデータを同じ領域に連続して並べる。例えば文字列情報を集めた場合、英語テキストならば、その文字コードは“127”(07FH:Hは16進数)未成の値をとる頻度が高く、文字の並びの相間も高いものとなる。また、日本語JISコードの場合にも同様に、上位バイト、下位バイトとも“0”～“255”的範囲内でまんべんなくコードが分散しておらず、ある範囲内の値しか取り得ないため、その文字コードの圧縮効果が期待できる。また、アドレス情報は、例えばアドレスの内部表現が32ビット長であつて、実装されているメモリが15Mバイト(=2の24乗)の場合、連続アドレスであれば最上位バイトは常に一定値であることが期待でき、同じ値が周期的に発生することが期待できる。このような場合は、高い圧縮効率が期待できる。

【0016】図6は、データ属性に対応した各バイトごとのデータ出現確率を示す図で、横軸は出現確率を示し、横軸はバイト値を示している。尚、これら図6

(a)～(i)において、横軸のスケールは同じではない。ここで、出現確率が一様である程、圧縮効率が期待できないないと考えられる。

【0017】図6 (e)はJISコードの各バイトの値の出現確率を示す図である。JISコードの下位バイトは、16進数で“21H”～“07EH”(Hは16進数を表す)の範囲に限定して分布し、上位バイトも“21H”～“74H”までで、かつ上位バイトの値が“50H”以上は、第2水準の漢字に相当しているため、その出現頻度が低くなっている。また、該当するJISコードが存在しない空き領域もあるので、このような分布

となる。

【0018】図6 (b) は、16ビット、32ビット整数の各ペイトの値の出現確率を示す図である。一般的な整数であるため、その出現頻度にほとんど差が認められない。しかし実際には、プログラムの性質に応じて出現頻度に差が生じると考えられる。また、16ビットデータが上詰めで格納されず、32ビットワードとして配置されている場合には、データによっては上位ペイトの空き領域に格納されるデータの出現頻度が変化することも考えられる。

【0019】図6 (c) はアドレス情報の最上位ペイトの値の出現確率を示している。ただし、本実施例の後述する図2のようなメモリマップの場合の分布である。

【0020】図6 (d) はアドレス情報の次位ペイトの出現確率を示している。

【0021】図6 (e) はアドレス情報の第3位のペイト値の出現確率を示している。

【0022】図6 (f) はアドレス情報の最下位のペイト値の出現確率を示している。この最下位ペイトでは、32ビット単位のデータが多いので数値が毎に周期的に変動する。尚、これらの図において、正確に図示しきれないもので、周期、間隔は正確ではない。

【0023】このように本実施例は、データの種類に応じて、その数値分布（出現確率）が異なっているので、それらデータをその属性毎に分離し、それらデータの出現確率を利用してデータを圧縮することにより、より効率良くデータを圧縮することを目的とする。

【0024】以下、本発明の一実施例を詳細に説明する。

【0025】図2は、本実施例のプリンタ装置200の構成を示すプロック図である。

【0026】201はホストコンピュータで、実施例のプリンタ装置200にページ記述言語等で記述された印刷コマンドや各種データを送出して印刷を実行させている。210は入出力部で、ホストコンピュータ201とプリンタ装置200との間のデータの送受信を制御している。この入出力部210は、例えばホストコンピュータ201より送信される、ホストスクリプトやページ記述言語等の印刷コマンドに応じたインターフェース部を有していても良い。211はCPUで、実施例のプリンタ装置200全体の動作を制御している。212は主メモリのRAM部分、213はそのROM部分で、そのデータ構成は後述の図1を参照して説明する。214はエンジン部で、例えばレーザビームプリント法、インクジェット法、サーマルプリント法等は熱転写法などにより、CPU211より送られる印刷イメージに応じて記録紙等の記録媒体に画像を印刷している。

【0027】以上の構成において、ROM213部分には、ホストコンピュータ201より受信した印刷コマンドに応じて実行される各アプリケーション・プログラム

及びフォントデータや各種データが記憶されている。CPU211は、ホストコンピュータ201より受信した印刷コマンドに応じて実行するアプリケーションを選択し、そのアプリケーションを使用して、その印刷コマンドの解釈、イメージへの展開などを実行している。またRAM212には、各アプリケーションの実行に応じて使用される作業領域（ワークエリア）が割り当てられており、CPU211は、そのアプリケーションの切換えに応じて使用するワークエリアを変更している。そのアプリケーションの切換えの際、使用しないアプリケーションのワークエリアのデータを圧縮することにより、自由に使用できるRAM212のワークエリアのメモリ空間を増やして、そのアプリケーションの実行を、より容易にしている。

【0028】図1は、複数のアプリケーションを切り換えて、ホストコンピュータ201から受け取った異なるコード体系のデータを印刷出力するプリンタ装置200におけるRAM212及びROM213のメモリマップ例を示す図である。

【0029】このメモリマップは、RAM212のデータ圧縮を全く行なうことなく、複数のアプリケーションのワークエリアをメモリに常駐させる場合で示している。各アプリケーションの作業エリアのうち、ページメモリ、受信データのバッファリング、処理途中の中間データの格納エリア等は共通に使用されるためRAM212上にとられ、アプリケーションの内部設定等を記憶するエリアは個別に割り当てられている。

【0030】図1において、213はROMエリアで、8Mペイトのメモリ空間を有している。100がアプリケーションAのコード領域（プログラム領域）、101がアプリケーションBのコード領域、そして102がアプリケーションCのコード領域である。103は、フォントデータの格納領域である。104は、アプリケーションに依存しない共通コード領域である。

【0031】RAM212の構成を説明すると、2はページメモリのイメージ展開領域を示し、3はフリーエリアで、受信データのバッファリング、中間データの格納等に使用される。イメージ展開領域2及びフリーエリア3は、各アプリケーション間で共有して使用されるが、異なったアプリケーションが起動される毎に初期化される。一般に、フリーエリア3のメモリ空間が広い程、より多くのデータをバッファリングできる。これにより、ホストコンピュータ201を速く、より印刷データの転送処理より解放することができたり、より多くのフォントデータをキャッシュして、印刷出力を高速化できる。

【0032】また、11はアプリケーションAの作業領域（ワークエリア）、12はアプリケーションBの作業領域、そして13はアプリケーションCの作業領域を示している。図1ではこれらアプリケーションの作業領域

11～13のメモリ空間のトータルエリアは、RAMエリア全体の5Mバイト(6000000H番地)のうち約2.2Mバイト(約2000000H)となっている。また、フリーエリア3のメモリ空間は300Kバイト程度である。

【0033】図3は、例えばアプリケーションCが起動され、それ以外のアクティブでないアプリケーション(アプリケーションA、B)のワークエリア11、12が圧縮された状態におけるメモリマップの内容を示す図である。

【0034】ここではアプリケーションCのみがアクティブになっており、11～1、12～1のそれぞれは、圧縮されたアプリケーションA、Bの作業領域を示しており、圧縮されていない状態(図1の作業領域11、12)のメモリ空間の3割程度にまで圧縮されている。

【0035】図3に示すように、各アプリケーションの作業領域11及び12が圧縮されることにより、図1の場合に比べて、RAM212に少なくとも約1Mバイトのワークエリアが確保されている。もしも、このメモリ(RAM212)の最終アドレスが80600000H番地で、RAM212のメモリ空間が5Mバイトだった場合には、図3に示す場合には、約1.3Mバイトの自由な使用できる余分のメモリエリアが確保できる。これにより、自由に使用できるワークエリアが増えたため、そのアプリケーションCの実行時に、そのエリアをフォントキッシュエリアの拡大等に使用できるため、作業性の向上が期待でき、印刷速度の向上が期待できる。

【0036】尚、本実施例では、図の簡易化のためにアプリケーションが3つの場合で示しているが本発明はこれに限定されるものではない。また、実際には各アプリケーション専用の作業領域はもっと狭いことが多いが、その代わりアプリケーションの数は多くなり、図示した程度の作業領域が必要となる。

【0037】次に、このようなデータ圧縮を行う際、そのデータの属性に基づいて圧縮する方法を説明する。

【0038】図4は、C言語等で作られたデータ構造を説明する図で、図4(a)は、双向リストと、データへのポインタ及びいくつかの整数や文字列等の定義例を示している。ここでは、「date」はキャラクタタイプで、「count」は32ビット整数、「left」及び「right」は、アドレスに定義されている。また、「flag」は16ビット整数として定義されている。

【0039】これらは処理系にも依存するが、例えばある32ビットの処理系では、メモリ上では図3(b)に示すように格納される。このままの配置では、「left」と「right」は共にアドレスであるためデータの相関があるが、「date」と「count」間、「count」と「flag」間、「flag」と「left」間においては、データ同士に何の相関も存在しない。

ない。またこれらの前後の変数と、「date」や「right」の属性が一致する可能性も低い。

【0040】前述したように、データを圧縮する際、同じ構造体のデータが連続して並んでいる場合は圧縮率が高まり、より高い圧縮率が期待できるが、このように互いに相間の無い配列の場合は、より高い圧縮効果は期待できない。

【0041】従つて本実施例では、これらのデータをそれぞれの属性に分けて格納することにより、後の圧縮処理において、より圧縮効果を高めている。

【0042】即ち、図4(b)において、「date」、「left」、「right」をいずれも32ビットのアドレス情報をとして扱う。また「count」を32ビットの整数、「flag」を16ビットの整数として、図4(c)のように同じ属性のデータを連続して格納する。

【0043】本実施例の図1及び図3に示すようなメモリ上のデータ構成の場合、アドレス情報を15進数で表示した場合、そのアドレスの取り得る範囲は、000000000H～000710000H、または80000000H～8051111FHの範囲に限定される。また、自分の領域以外のアドレスを指すことはないので、このアドレスの範囲はさらに限定される。この場合、最高位バイトの値は“00”または“80”となり、次位バイトの値は最高位バイトの値が“00”ならば“23”～“7F”的値を取り得、“80”ならば“00”～“30”的値を取り得る。

【0044】このように本実施例によれば、このメモリ空間をアクセスするアドレス値に高い相関が認められる。また最低位バイトは、文字列のポインタを除き、下位2ビットは常に“0”(4バイト単位でアクセスする)であることから、このアドレスもまた高い圧縮効果を期待できる。

【0045】例えばLZW圧縮等において、アドレス情報と他の情報がランダムに混在している場合と、アドレス情報を分離した場合とで比較すると、そのアドレスを分離した場合は、数パーセントの圧縮率の向上が見られる。

【0046】以上の構成に基づく本実施例のプリンタ装置200における処理を、図5のフローチャートを参照して説明する。尚、この処理を実行する制御プログラムは、ROM213の共通コードエリア104に記憶されている。

【0047】図5のフローチャートで示す処理は装置の電源が投入されることにより開始され、まずステップ1で、全体の初期化設定を行なって各アプリケーションのイニシャライズを行ない、ステップS2で、デフォルトで設定されているアプリケーションをアクティブにして、ポストコンピュータ201よりの印刷データの受信待ちに入る。

【0048】ステップS3で、ホストコンピュータ201よりのデータを受信するとステップS4に進み、その受信データ或は受信したインターフェースに応じて、実行するアプリケーションを選択する。こうして選択されたアプリケーションが現在アクティブなアプリケーションかどうかを判断し、そうであればステップS7に進み、そのまま対応するアプリケーションのコード100ないし102のいずれかの領域のプログラムに制御を移行し、ホストコンピュータ201よりのデータを受信し（ステップS7）、その印刷コマンドを解釈して印刷イメージに展開して印刷を行なう（ステップS8）。こうしてステップS9で、ホストコンピュータ201より受信した所定量の印刷（1つの印刷ジョブ）が終了するとステップS3に戻り、次の印刷コマンドの受信待ちに進む。

【0049】一方、ホストコンピュータ201より受信した印刷コマンドにより選択されたアプリケーションが、その時点で非アクティブであった場合にはステップS4よりステップS5に進み、その時点でアクティブであつたアプリケーションの作業領域（ワークエリヤ）のデータを圧縮して退避する。そして、その印刷コマンドにより要求されたアプリケーションの作業領域を展開し、該当アプリケーションに制御を移行する。次にステップS5に進み、イメージ展開領域2、フリーエリヤ3の共有データ領域を初期化する。こうして新たなアプリケーションが起動されると、ホストコンピュータより送られてくる印刷データを受信して解釈し（ステップS7）、印刷出力を行なう（ステップS8）。こうして、そのアプリケーションに対応した印刷ジョブに対応する印刷処理を終るとステップS3に戻り、次の印刷コマンドの受信待ちに進む。

【0050】ここで、ステップS5において、RAM'212の各アプリケーションのワークエリヤでは、例えば前述の図4に示すように、データの属性が一致するデータが連続して記憶されているので、よりデータの圧縮効率を高めることができる。

【0051】尚、本発明は、複数の機器から構成されるシステムに適用しても1つの機器から成る装置に適用しても良い。また、本発明は、システム或は装置に本発明を実施するプログラムを供給することによって達成される場合にも適用できる。

【0052】以上説明したように本実施例によれば、複数のアプリケーションを切り換えて使用するようなプリンタ装置において、アクティブになつてないアプリケーションの作業領域を圧縮する際に、本実施例のようにな、属性が一致するデータ毎に連続してメモリに格納することにより、データの圧縮効率を高めて、より効率的にメモリを使用することができる。

【0053】

【発明の効果】以上説明したように本発明によれば、メ

モリ上のデータを効率良く圧縮できる効果がある。

【0054】また本発明によれば、外部機器よりの印刷コマンドに応じて、印刷装置内で実行するアプリケーションを高速に切り換えて印刷を行うことができる。

【0055】また本発明によれば、データを効率良く圧縮して、アプリケーションを実行するためのワークエリヤを十分に確保できる効果がある。

【0056】

#### 【図面の簡単な説明】

【図1】本実施例のプリンタ装置において、データ圧縮を行なわないときのメモリマップを示す図である。

【図2】本実施例のプリンタ装置の概略構成を示すプロック図である。

【図3】本実施例のプリンタ装置において、アクティブでないアプリケーションに関するデータを圧縮したときのメモリマップを示す図である。

【図4】C言語により規定されたデータの一例を説明するための図で、(a)はデータ構造を定義するC言語の一例を示す図、(b)はC言語で規定されたデータ構造がどのように配置されるかを示す図、(c)は本実施例によりデータの属性に応じて分離されて配置されたデータ配列を示す図である。

【図5】本実施例のプリンタ装置における処理を示すフローチャートである。

【図6】データに応じた出現確率を示す図で、(a)はJISコードの各バイトの値の分布確率を示す図。

(b)は16ビット、32ビット整数の各バイトの値の分布確率を示す図、(c)は、本実施例のアドレス情報の最高位バイトの値の分布確率を示す図、(d)は本実施例のアドレス情報の次位バイトの分布確率を示す図、(e)は、本実施例のアドレス情報の第3位バイトの分布確率を示す図、(f)はアドレス情報の最下位バイトの値の分布確率を示す図である。

#### 【符号の説明】

2 イメージ展開領域

3 フリーエリヤ

1 1 アプリケーションAの作業領域

1 2 アプリケーションBの作業領域

1 1-1 圧縮されたアプリケーションAの作業領域

1 2-1 圧縮されたアプリケーションBの作業領域

1 3 アプリケーションCの作業領域

2 0 アプリケーションに依存しない共通作業領域

1 0 0 アプリケーションAのコード領域

1 0 1 アプリケーションBのコード領域

1 0 2 アプリケーションCのコード領域

1 0 3 フォントデータ格納領域

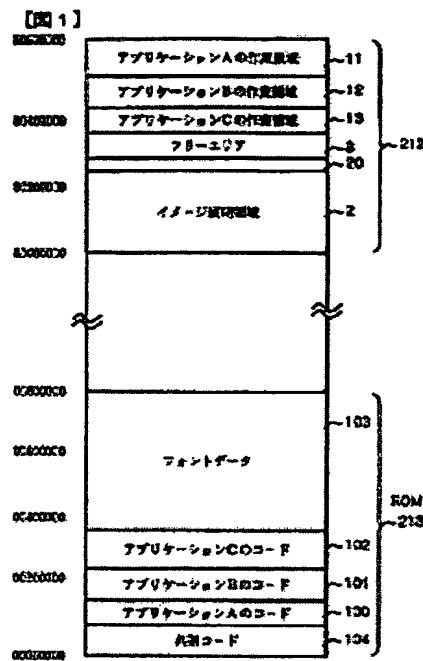
1 0 4 共用コード領域

2 0 0 プリンタ装置

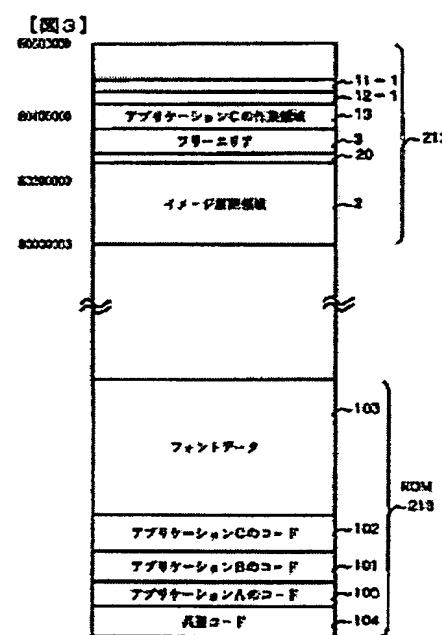
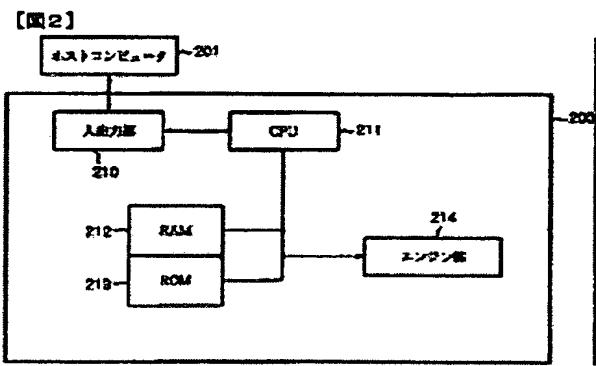
2 0 1 ホストコンピュータ

2 1 1 CPU

212 ROM



213 RAM



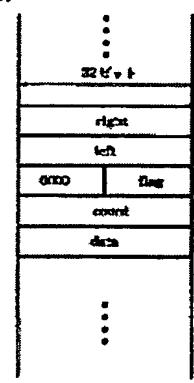
[図4]

```

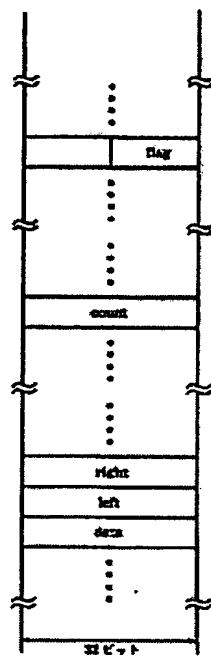
typedef struct node {
    char *data;
    int count;
    struct node *left;
    struct node *right;
} node;

```

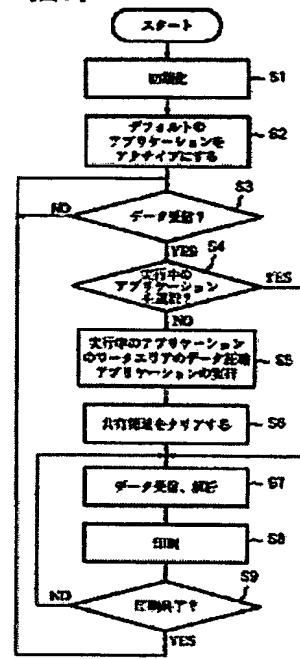
(b)



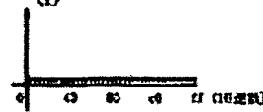
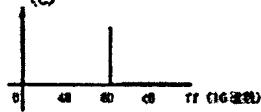
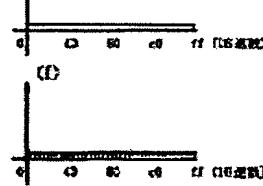
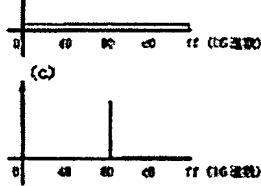
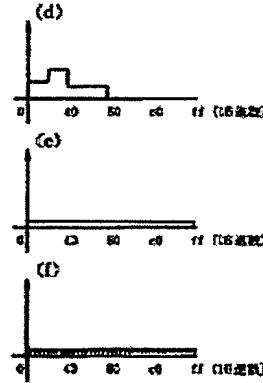
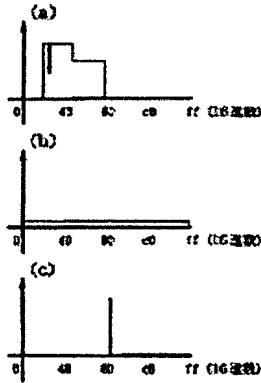
(c)



[図5]



[図6]



**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**